# VI & VIC

Nr. 4 - 2. årg. - August-september 1983 Pris 17,95

# Programmet kan føre din kassebog - og det holder også regnskab med momsen




Der findes naturligvis allerede færdigudviklede økonomiprogrammer, men interessen for denne type arbejdsredskaber til VIC-20 er stor - og her er derfor et program, som kan føre en kassebog, et momsregnskab og fordele posterne på tre forskellige produkter - og på ti forskellige posteringstyper (tre slags indtægter og syv slags udgifter). Hvis brugeren har et andet behov, må man dykke ned i programmet og ændre det på passende steder.

En kassebog eller et momsregnskab, der ikke kan komme ned på papir, har man ikke så megen glæde af, og programmet er derfor skrevet, så det forudsætter en printer - men det fordrer ikke nogen diskettestation.

En disketteversion vil naturligvis arbejde hurtigere, men mon ikke brugere med et begrænset bogføringsbehov kan leve med at skulle hente program og data fra kassettebånd. Altså, nødvendigt hardware: VIC-20 med en kraftig RAM-udvidelse, båndstation, printer.

Hvis man ikke orker at taste programmet ind, sender vi gerne en kopi på kassettebånd til de sædvanlige 20 kr. - som til gengæld må vedlægges bestillingen i check eller frimærker.

### Instruktioner på skærm

Herefter skulle der egentlig ikke være flere problemer. Efter at have fortalt computeren, hvilken dato, man befinder sig på, opstilles en menu med syv forskellige muligheder, nemlig:

A INPUT
B SØGE
C DATA FRA TAPE
D DATA TIL TAPE
E SKRIVE KASSEBOG
F MOMSREGNSKAB
G RESULTATBEREGNING

Hver for sig fører disse hen til en række operationer, som ikke skulle give nogen vanskeligheder. Måske bør det nævnes her, at man i SØGE-funktionen kan skrive enten et bilagsnummer, hvorefter indholdet af det pågældende nummer dukker op på skærmen. Ønsker man ikke at se et bestemt nummer, kan man blot trykke RETURN, hvorefter computeren spørger efter en bestemt tekst - og her kan man skrive så få eller mange bogstaver, man mener er nødvendigt at bruge for at finde den relevante post.

Skriver man blot et enkelt bogstav og trykker RETURN, vil man få alle de bi-

*Fortsættes side 2*

## »Kvik-find« program til Commodore 64

Læs side 6

## Ny bog om Commodore BASIC

Læs side 12

## Mere om maskinsprog på VIC

Læs side 13

*Fortsat fra forsiden*

lag frem, der har en tekst, der begynder med det indskrevne bogstav. Man kommer videre ved at trykke på en hvilken som helst tast - og trykker man på pil-til-venstre tasten, vender man straks tilbage til menuen.

## Med og uden moms

Indtastningerne foretages med beløb inkl. moms, hvorefter der gøres rede for, om der er moms i beløbet eller ej. Når kassebogen skrives ud, er der ligeledes tale om bruttobeløb, men når man (under funktion G) får udskrevet beregninger af resultatet, arbejdes der med nettobeløb, altså uden moms.

## MID$

Kunsten med at få skilt alle disse data fra hinanden udføres med MID$ og LEFT$. Se i brugervejledningen side 131 og 132.
Kort fortalt virker disse funktioner sådan, at man kan dele strenge. En streng i vort system kan f.eks. se således ud:

"21/05ABJTEKST TIL POSTEN"

Strengens navn kan f.eks. være PO$(21), og hvis man skal have fat i oplysning om, hvilket produkt, den vedrører, hedder instruktionen MID$(PO$(21),7,1), som vil give os tegnet B. Skiftes 7-tallet ud med 8, får vi i stedet et "J", som er en oplysning om, at posten indeholder moms.

## Sådan bruges programmet

Når REGNSKAB skal tages i brug, må man allerførst overveje sit eget behov. Programmet er fra "fødslen" indrettet til at arbejde med tre kategorier af indtægter og syv forskellige muligheder for at postere udgifter. Der er gjort plads til tre forskellige aktiviteter eller produkter. Kan man uden videre bruge denne konstruktion, må man blot dykke ned i datasætningerne i linie 100 til 200 og dér skrive sine egne betegnelser. Har man brug for færre muligheder, kan man lade resten stå hen - men skal systemet udvides på den ene eller anden led, må man gå i gang med programmeringen.

## Strenge

Princippet i programmet er, at oplysningerne gemmes i to tabeller - én med beløbene for de enkelte posteringer og én med streng-variabler, der indeholder oplysninger om dato for bogføringen, om hvilket produkt, posten vedrører, om hvilken udgift eller indtægt det er, om beløbet indeholder moms, der er fradragsberettiget - og endelig en tekst til det enkelte bilag.
Lad os se lidt nærmere på strengene. Hver streng, der har navnet PO$(X), hvor X er streng-variablens nummer. I programmet er der gjort plads til 500 poster.

De første fem karakterer i hver streng betegner datoen, og derfor er det vigtigt, at der altid benyttes fem tegn, når datoen skal indtastes, f.eks. "21/05" for den 21. maj - men der er lagt en lille kontrol ind, så datoen virkelig får sine fem cifre.

Næste karakter, den sjette, betegner hvilken udgift eller indtægt, der er tale om, mens den syvende karakter er en kode for hvilket produkt, der er tale om. Herefter følger en karakter (nummer otte), der fortæller, om der er moms i beløbet, og de resterende pladser er så den tekst, der betegner posteringen i regnskabet.

# SUK

... mon man kunne lokke jer til at viderebringe følgende hjertesuk i næste nummer af bladet?

Er der ikke nogen, VIC-ejere eller forhandlere, som kunne tænke sig at oprette en form for »printerservice« for fattiglus, så man kunne få lavet en liste af sine møjsommeligt indtastede programmer - der er ikke noget så ærgerligt, som at få loadet Kaj og Andrea ind til en »load error« og derefter opdage, at der er salat i reservetapen og leverpostej på den vigtigste linie i håndlisten (som for øvrigt er ulæselig og ikke rettet efter diverse syntax error'er).

- - -

Hjertesukket kommer fra Bent R. Husted, og vi bringer det gerne videre. Måske var der også VIC-folk, der kunne hjælpe hinanden med at transformere VIC-programmer til Commodore 64 . . ?




VI & VIC

**VI & VIC udgives af commodore data a/s, Bjerrevej 67, 8700 Horsens.**
**Redaktion og tilrettelæggelse:**
**Baghuset Grafik/Design, Vejle.**
**Ansvarshavende redaktør: Steen Venbjerg.**
**Annoncer: Lise Tylvad,**
**Æblehaven 7, 7100 Vejle.**
**Tryk: Baghuset Grafik/Design, Vejle.**
**Redaktion og ekspedition:**
**Kornager 29, 7100 Vejle.**
**Giro nr. 6 17 39 93**


ISSN 0108-2213

# VI & VIC's regnskabsprogram kan du også få på kassettebånd

```
0 S1=36876:S2=36875
1 S3=36874:VO=36878:ST$="    ****************":US$="─────────────────────":N=500
2 PRINT"□■HVILKEN DATO            ER DET I DAG":GOSUB10:INPUTDD$
4 IN$="INDTÆGTER I ALT":UD$="UDGIFTER I ALT"
5 DEFFNBE(X)=(ASC(MID$(PO$(X),6)))-64
6 DEFFNPR(X)=(ASC(MID$(PO$(X),7)))-64
7 DEFFNA(X)=INT(X/1.22*100+.5)/100:DEFFNB(X)=X-FNA(X)
8 DEFFNC(X)=INT(X*.22*100+.5)/100:DEFFND(X)=X+FNC(X):FORT=1TO15:CL$=CL$+"
 ":NEXT
9 DIMR(10,3),PO(N),PO$(N):GOTO1000
10 POKEVO,5:POKES1,222:FORD=1TO60:NEXT:POKES1,0:RETURN
11 POKEVO,15:POKES3,211:FORT=1TO99:NEXT:POKES3,0:FORT=1TO111:NEXT:POKES3,144:FOR
T=1TO99:NEXT:POKES3,0:RETURN
12 GETS$:IFS$=""THEN12
13 RETURN
15 PRINT"■  ***  RIGTIGT ? ***□":GOSUB10:GOSUB12:PRINT"                          ":RE
TURN
18 PRINT"■  ***  MERE NU ? ***□":GOSUB10:GOSUB12:PRINT"                          ":RE
TURN
20 POKE251,PEEK(55):POKE252,PEEK(56):POKE55,PEEK(51):POKE56,PEEK(52):X=FRE(8):RE
TURN
21 POKE55,PEEK(251):POKE56,PEEK(252):RETURN
24 PRINT" ** PRINTER KLAR ? **□":GOSUB10:GOSUB12:PRINT"                         ":GO
SUB10:RETURN
30 REM** KONTROL ***
35 PRINT"□■BILAG NR.■"BI+V:PRINT"■■DATO:■ ":PRINTLEFT$(PO$(V),5):PRINT"■■VEDRØRE
NDE■"
36 RESTORE:FORT=1TOASC(MID$(PO$(V),6))-64:READA$:NEXT:PRINTA$
37 RESTORE:FORT=1TO10:READA$:NEXT:FORT=1TOASC(MID$(PO$(V),7))-64:READA$:NEXT:PRI
NTA$
55 PRINT"■■TEKST:■ ":PRINTMID$(PO$(V),9):PRINT"■■BELØB:■ "PO(V)"KR."
56 IFMID$(PO$(V),8,1)="J"THENPRINT"■NETTO    "FNA(PO(V)):PRINT"MOMS     "FNB(PO(V)
)
59 RETURN
80 PRINT"□■■  *** PAPIR OK? ***":GOSUB10:GOSUB12:LI=0:GOSUB81:RETURN
81 PRINT"□■■■■■"ST$:PRINT"■            VENT■":PRINTST$:RETURN
85 PRINT#4,CHR$(14)CHR$(16)"66SIDE"SI
87 PRINT#4,CHR$(15)"DATO  NUMMER  TEKST"CHR$(14)CHR$(16)"52UD  IND  SALDO"
88 PRINT#4,CHR$(15):RETURN
90 FORT=1TO5:PRINT#4,CHR$(10):NEXT:PRINT#4,"MED VENLIG HILSEN"CHR$(10)
91 PRINT#4,CHR$(10)"STEEN VENBJERG":RETURN
95 PRINT#4,"BETALINGSBETINGELSER: 30 DAGE FRA FAKTURADATO.":RETURN
100 DATAINDTÆGT 1,INDTÆGT 2,INDTÆGT 3
110 DATAUDGIFT 1,UDGIFT 2,UDGIFT 3,UDGIFT 4,UDGIFT 5, UDGIFT 6,UDGIFT 7
200 DATAPRODUKT 1,PRODUKT 2,PRODUKT 3
701 PR=INT(PR*100+.5)/100:PR$=STR$(PR):IFPR<1ANDPR>0THENPR$="0"+MID$(PR$,2):GOTO
704
702 IFPR-INT(PR)=0THENPR$=PR$+"."
703 IFPR<0ANDPR>-1THENPR$="-0"+MID$(PR$,2)
704 IFASC(RIGHT$(PR$,3))<>46THENPR$=PR$+"0":GOTO704
815 HL$=LEFT$(TR$+CL$,LL-LEN(PR$))+PR$:RETURN
1000 REM***START PROGRAM***
1010 PRINT"□■"ST$:PRINTTAB(7)"■REGNSKAB■":PRINTST$
1020 PRINT"■ A  INPUT":PRINT"■ B  SØGE":PRINT"■ C  DATA FRA TAPE"
1030 PRINT"■ D  DATA TIL TAPE":PRINT"■ E  SKRIVE KASSEBOG"
1040 PRINT"■ F  MOMSREGNSKAB":PRINT"■ G  RESULTAT"
1100 GOSUB10:GOSUB12:IFS$<"A"ORS$>"G"THENGOSUB11:GOTO1100
1110 S=ASC(S$)-64:ONSGOTO1200,2000,3000,1600,4000,5000,6000
```

```
1200 REM***INPUT***
1220 PRINT"♥"TAB(8)"INPUT":PRINTUS$:IFBI=0THENGOSUB10:INPUT"SIDSTE BILAG NR.";BI
1223 IFMA$=""THENINPUT"▪HVILKET AAR";MA$
1224 IFGS=0THENINPUT"▪GAMMEL SALDO";GS
1225 FORV=1TON:IFPO$(V)<>""THEN1590
1232 PRINT"♥▪BILAG NR.▪"BI+V:PRINT
1233 INPUT"DATO";I$(1)
1234 IFLEN(I$(1))<>5THENGOSUB11:PRINT"▪5 CIFRE - TAK!▪":GOTO1233
1240 INPUT"▪▪TEKST";I$(5):PRINT"♥▪▪BESKRIV POSTEN":PRINT"▪▪INDTAEGTER:▪"
1241 RESTORE:FORT=1TO3:READA$:PRINT" "CHR$(T+64)" "A$:NEXT:PRINT"▪UDGIFTER:▪"
1242 FORT=4TO10:READA$:PRINT" "CHR$(T+64)" "A$:NEXT
1245 GOSUB12:IFS$<"A"ORS$>"J"THENGOSUB11:GOTO1245
1250 RESTORE:FORT=1TOASC(S$)-64:READA$:NEXT:PRINT"♥▪▪"A$:I$(2)=S$
1300 RESTORE:FORT=1TO10:READA$:NEXT
1341 PRINT"▪PRODUKT?▪":FORT=1TO3:READA$:PRINT" "CHR$(T+64)" "A$:NEXT
1345 GOSUB12:IFS$<"A"ORS$>"C"THENGOSUB11:GOTO1345
1350 RESTORE:FORT=-9TOASC(S$)-64:READA$:NEXT:PRINT"▪"I$(5):PRINT"▪▪▪"A$;CL$:I$(3
)=S$
1360 GOSUB10:INPUT"▪BELØB";PO(V)
1370 IFLEFT$(I$(5),4)="MOMS"THENFORT=1TO10:GOSUB10:NEXT:PRINT"MOMSBETALING?":GOS
UB12:IFS$="J"THENS$="X"
1371 IFS$="X"THEN1410
1401 GOSUB10:PRINT"MOMS-EFFEKT?":GOSUB12:IFS$<>"J"ANDS$<>"N"THENPRINT"SKRIV J EL
LER N":GOSUB11:GOTO1401
1410 I$(4)=S$:PO$(V)="":FORT=1TO5:PO$(V)=PO$(V)+I$(T):NEXT
1420 GOSUB30:GOSUB15:IFS$<>CHR$(13)THEN1232
1580 M=V:GOSUB18:IFS$<>CHR$(13)THEN1000
1590 NEXT
1591 GOSUB11:PRINT"IKKE MERE PLADS":GOSUB16:GOTO1000
1600 REM *** DATA TIL TAPE
1601 PRINT"♥":OPEN1,1,1:GOSUB81
1610 PRINT#1,M:PRINT#1,BI:PRINT#1,GS:PRINT#1,MI:PRINT#1,MO:PRINT#1,MA$
1620 FORT=1TOM:PRINT#1,PO$(T):PRINT#1,PO(T)
1630 NEXT:CLOSE1:PRINT"♥▪SKAL DATA TJEKKES?":GOSUB10:GOSUB12:IFS$="N"THEN1000
1640 GOTO9000
2000 REM**GENNEMSE***
2010 X=0:INPUT"♥▪ET BESTEMT NUMMER";X:IFX=0THEN2020
2015 V=X-BI:IFV<1ORV>MTHENGOSUB11:GOTO2010
2019 GOSUB30:GOSUB12:GOTO1000
2020 X$="":INPUT"EN BESTEMT TEKST";X$
2100 FORV=1TOM:IFPO$(V)=""ORX$<>MID$(PO$(V),9,LEN(X$))THEN2190
2110 GOSUB30:GOSUB12:IFS$="←"THENV=M
2190 NEXT:GOTO1000
3000 REM *** DATA FRA TAPE
3001 GOSUB10:PRINT"♥":OPEN1,1,0:GOSUB81
3010 INPUT#1,M:INPUT#1,BI:INPUT#1,GS:INPUT#1,MI:INPUT#1,MO:INPUT#1,MA$
3015 FORV=1TON:PO$(V)="":PO(V)=0:NEXT
3020 FORT=1TOM:INPUT#1,PO$(T):INPUT#1,PO(T)
3030 NEXT:CLOSE1:GOTO1000
4000 REM***SKRIVE KASSEBOG***
4010 GOSUB10:PRINT"♥":GOSUB24:GOSUB80:OPEN4,4:GOSUB81:UD=0:IN=0:SI=1
4020 PRINT#4,CHR$(14)"BEVAGELSER I "MA$:PRINT#4,CHR$(15)"UDSKREVET "DD$:PRINT#4:
GOSUB85
4035 SA=GS:TR$="                   OVERFØRT":PR=GS:LL=80:GOSUB701:PRINT#4,HL$
4036 GOSUB20
4050 FORV=1TON:IFPO$(V)=""THENV=N:GOTO4090
4055 LL=56:IFMID$(PO$(V),6,1)<"D"THENLL=66
4060 TR$=LEFT$(PO$(V),5)+STR$(V+BI)+"      "+MID$(PO$(V),9):PR=PO(V):GOSUB701:PR=
0:TR$=HL$
4070 IFLL=56THENSA=SA-PO(V):UD=UD+PO(V):GOTO4080
4075 SA=SA+PO(V):IN=IN+PO(V)
4080 LL=80:TR$=HL$:PR=SA:GOSUB701:PRINT#4,HL$:LI=LI+1:IFLI=55THENSI=SI+1:GOSUB80
:GOSUB85
4090 NEXT:PRINT#4
4091 TR$="I ALT":PR=UD:LL=56:GOSUB701:TR$=HL$:LL=66:PR=IN:GOSUB701:TR$=HL$:LL=80
:PR=SA
4092 GOSUB701:PRINT#4,HL$
4099 CLOSE4:GOTO1000
5000 REM***SKRIVE MOMSREGNSKAB***
5005 PRINT"♥":GOSUB24:
5010 IFG2=0THENINPUT"GL. SUM - UD";G2
5011 IFG1=0THENINPUT"GL. SUM - IND";G1
5012 MI=G1:MO=G2:OPEN4,4:LI=0
5015 PRINT"HVILKET KVARTAL?"
```

```
5016 GOSUB12:SM=3*VAL(S$):FM=SM-2:IFS$<"1"ORS$>"4"THENGOSUB11:GOTO5016
5020 PRINT#4,CHR$(14)"MOMSREGNSKAB I "MA$CHR$(15)", "SM/3". KVARTAL"CHR$(15)
5021 PRINT#4,"UDSKREVET "DD$:SI=1
5030 GOSUB85
5033 SA=MI-MO:GOSUB20
5035 TR$="                    OVERFØRT":PR=MO:LL=56:GOSUB701
5036 TR$=HL$:PR=MI:LL=66:GOSUB701:TR$=HL$:PR=SA:LL=80:GOSUB701:PRINT#4,HL$
5040 FORV=1TON:IFPO$(V)=""THENV=N:GOTO5090
5050 MN=VAL(MID$(PO$(V),4,2)):IFMN<FMORMN>SMTHEN5090
5051 X=.1803:LL=66:IFMID$(PO$(V),8,1)="N"THEN5090
5052 IFMID$(PO$(V),8,1)="X"THENX=1
5055 IFMID$(PO$(V),6,1)>"C"THENLL=56:SA=SA-PO(V)*X:MO=MO+PO(V)*X:GOTO5060
5056 SA=SA+PO(V)*X:MI=MI+PO(V)*X
5060 TR$=LEFT$(PO$(V),5)+"  "+STR$(BI+V)+"      "+MID$(PO$(V),9):PR=PO(V)*X
5070 GOSUB701:TR$=HL$:PR=SA:LL=80:GOSUB701:PRINT#4,HL$
5080 LI=LI+1:IFLI=55THENSI=SI+1:GOSUB80
5090 NEXT
5091 TR$="I ALT":PR=MO:LL=56:GOSUB701:TR$=HL$:LL=66:PR=MI:GOSUB701:TR$=HL$:LL=80
:PR=SA
5092 GOSUB701:PRINT#4,CHR$(10)HL$
5099 GOSUB21:CLOSE4:GOTO1000
6000 REM***BEREGNING AF RES.***
6010 GOSUB81:FORB=1TO10:FORP=1TO3:R(B,P)=0:NEXTP,B
6020 FORV=1TOM
6030 IFMID$(PO$(V),8,1)="X"THEN6090
6040 IFMID$(PO$(V),8,1)="J"THENPL=FNA(PO(V)):GOTO6080
6050 PL=PO(V)
6080 R(FNBE(V),FNPR(V))=R(FNBE(V),FNPR(V))+PL
6090 NEXT
7000 REM***UDSKRIVNING AF RES.****
7001 FORT=1TO3:I(T)=0:NEXT:FORB=1TO3:FORP=1TO3:I(P)=I(P)+R(B,P):NEXTP,B
7002 FORT=1TO3:U(T)=0:NEXT:FORB=4TO10:FORP=1TO3:U(P)=U(P)+R(B,P):NEXTP,B
7010 GOSUB80:OPEN4,4:RESTORE
7020 FORT=1TO10:READA$:NEXT:GOSUB20
7030 PRINT#4,CHR$(14)"RESULTAT I "MA$CHR$(10)CHR$(15)
7035 PRINT#4,"BEREGNET "DD$CHR$(10)
7040 LL=32:TR$="":PR$="":FORT=1TO3:LL=LL+12:TR$=HL$:READPR$:GOSUB815:NEXT:TR$=HL
$
7050 PR$="I ALT":LL=LL+12:GOSUB815:PRINT#4,HL$CHR$(10):RESTORE
7090 PRINT#4,CHR$(14)"INDTAGTER"CHR$(10)CHR$(15)
8000 FORBE=1TO10:READHL$
8010 Z=0:LL=32:PR$="":FORT=1TO3:LL=LL+12:TR$=HL$:PR=R(BE,T):Z=Z+PR:GOSUB701:NEXT
8011 TR$=HL$:LL=LL+12:PR=Z:GOSUB701:PRINT#4,HL$
8020 IFBE=3THENZ=0:LL=32:HL$=IN$:FORT=1TO3:LL=LL+12:TR$=HL$:PR=I(T):Z=Z+PR:GOSUB
701:NEXT
8030 IFBE=3THENTR$=HL$:PR=Z:LL=LL+12:GOSUB701:PRINT#4,CHR$(10)HL$CHR$(10)CHR$(10
)
8035 IFBE=3THENPRINT#4,CHR$(14)"UDGIFTER"CHR$(10)CHR$(15)
8090 NEXT
8100 Z=0:PRINT#4:LL=32:HL$=UD$:FORT=1TO3:LL=LL+12:TR$=HL$:PR=U(T):Z=Z+PR:GOSUB70
1:NEXT
8110 LL=LL+12:TR$=HL$:PR=Z:GOSUB701:PRINT#4,HL$CHR$(10)
8120 Z=0:LL=32:HL$="RESULTAT":FORT=1TO3:LL=LL+12:TR$=HL$:PR=I(T)-U(T):Z=Z+PR:GOS
UB701:NEXT
8130 LL=LL+12:TR$=HL$:PR=Z:GOSUB701:PRINT#4,HL$CHR$(10)
8150 GOSUB21:CLOSE4:GOTO1000
8999 WAIT653,1:LIST6001-8998
9000 REM *** TEST AF DATA FRA TAPE
9001 PRINT"▯▮SPOL TILBAGE           OG TRYK ▮RETURN▀":GOSUB12
9005 PRINT"▯▮TRYK ▮STOP▀":WAIT37151,126,62:PRINT"▯"
9010 OPEN1,1,0:GOSUB81:INPUT#1,M:INPUT#1,BI:INPUT#1,GS:INPUT#1,G1:INPUT#1,G2:INP
UT#1,MA$
9020 FORT=1TOM:INPUT#1,PO$:INPUT#1,PO
9030 IFPO$<>PO$(T)ORPO<>PO(T)THENPRINT"▯▮FEJL":PRINT"VEDR. BILAG NR."T+BI:GOSUB1
1:GOSUB12:GOTO9090
9040 PRINT"███BILAG NR."BI+T:PRINT"█"MID$(PO$(T),9)" ▮ER OK▀":GOSUB10
9090 NEXT:CLOSE1:GOTO1000
63990 END
63998 OPEN1,8,15:SAVE"DUMMY",8:PRINT#1,"S0:REGNSKAB":PRINT#1,"R0:REGNSKAB=DUMMY"
:CLOSE1
63999 VERIFY"REGNSKAB",8

READY.
```

# Her er opskriften på et »kvik-find« program til Commodore 64

VIC-20 brugere kan nikke genkendende til dette lille program. Og du kan spørge en af dem, om de ikke har været glade for det. Derfor viser vi her Kvikfind i en Commodore 64 version.

Det mest irriterende ved at bruge en kassettebåndoptager som lager-medium er den ventetid, man har, indtil Commodore 64 har fundet frem til programmet, der måske er lagret i slutningen af et C-60 bånd.

Og hvem kender ikke den situation, hvor man har glemt at nulstille tælleren, eller er kommet i tvivl om man nu har husket det, når man er ved at spole frem til et program.

Disse og lignende situationer kan man godt glemme alt om, hvis man bruger Kvikfind som første program på alle sine kassettebånd.

Så snart programmet er indlæst i Commodore 64, og man har tastet RUN, får man en menu op på skærmen. Man vælger sit program ved at taste et tal ind, og fluks kommer ordren om at trykke F.FWD. Maskinen stopper automatisk ved det ønskede program, og er klar til indlæsning.

## Sådan gør du

1. Båndet med Kvikfind spoles helt tilbage.
2. LOAD
3. RUN
4. Vælg hvilket programnummer du ønsker.
5. Tryk F.FWD når C-64 ber' om det.
6. Tryk stop når C-64 siger til.
7. Tryk PLAY hvorefter programmet loades.

Fremgangsmåden er nøjagtig den samme som for det Kvikfind-program, vi bragte i det allerførste nummer af VI & VIC.

## Klar til optagelse

Først gør du dig klart, hvilke programmer du ønsker på båndet foruden Kvikfind. De skal nemlig navngives i datasætningen sidst i programmet. Der skal være overensstemmelse mellem programnavnene i datalinierne og i den titel, du hæfter på SAVE kommandoen senere.

Fremgangsmåden er følgende, når du skal indspille et nyt bånd med Kvikfind og et antal programmer:

1. LOAD (eller skriv) Kvikfind ind i Commodore 64.
2. Tag Kvikfind-båndet ud (master-båndet) og sæt en frisk kassette i. SAVE" Kvikfind*spil" f.eks. eller blot SAVE"Kvikfind". Spol IKKE tilbage. Pil kassetten ud.
3. LOAD det program du ønsker på position et fra et eksisterende bånd. Pil kassetten ud.
4. Sæt båndet med "Kvikfind*spil" i recorderen og SAVE"(navnet - der skal svare til data)".
5. Spol tilbage. LOAD og RUN.
6. Vælg hvilken position næste program skal SAVEes fra. Tryk tallet og RETURN. Når C-64 ber' om, at du trykker STOP, trykkes dernæst RUN/ STOP for at afbryde den normale brugerrytme.
7. Start fra punkt 3 igen.

## Tips og faldgrupper

Du kan lagre lige så mange programmer på båndet, der er plads til. Linie 500 indeholder variablen N, der angiver antallet af programmer. Men husk, at DATA i linierne 590 og fremefter skal svare til antallet af programmer.

Hvert program fylder seks sekunders fremspoling på båndet. Hvis du ikke tror det strækker, skal du gå ind i linie 570. 600 svarer til ti sekunder. Et sekund mere: 660 osv. Linie 570 er i øvrigt programmets ankermand. Det er her vi, ved hjælp af TI variablen, får udlæst digitaluret i C-64.

Du kan »snyde« computeren fra dens krav om samhørighed mellem DATA og titlerne på programmerne. I stedet for at skrive "program 1, osv" i DATA, kan du sætte N antal kommaer. Fordelen ved det er, at kassetten er lettere at lagre programmerne på. Skriv blot SAVE.

Til gengæld mister du informationerne om, hvilket bånd der indspilles, og når du LOADer, vil det være det næstfølgende program, som lagres i C-64.

I øvrigt henvises til de REM kommentarer, der er givet i programmet. Dem behøver du naturligvis ikke taste med ind.

```
500 N=7:DIMN$(N):FORI=1TON:READN$(I):NEXT
510 PRINT"■PROGRAM ■KVIKFIND■ NUMMER OG NAVN■"
520 FORI=1TON:PRINTI;"";N$(I):NEXTI
530 INPUT"■FIND NUMMER■";J:PRINT
540 IFJ<1ORJ>NTHEN510
550 IFJ=1THEN580
560 IFPEEK(1)<>55THENPRINT"■TRYK ■STOP■ PAA RECORDER"
561 IFPEEK(1)<>55THEN561
564 PRINT"■TRYK ■F.FWD■":SYS63524:A=TI
570 IFABS(TI-A)<(J-1)*600THEN570
575 PRINT"■TRYK ■STOP■ PAA RECORDER"
577 SYS64659:IFPEEK(1)<>55THEN577
580 PRINT"■";CHR$(17);CHR$(17);CHR$(17);"LOAD";CHR$(34);N$(J);CHR$(34);CHR$(19)
585 POKE198,1:POKE631,13:END
590 DATAPROGRAM 1,PROGRAM 2,PROGRAM 3,PROGRAM 4,PROGRAM 5
595 DATAPROGRAM 6,PROGRAM 7
```

# Flere måder at håndtere tal, der skal stå pænt under hinanden

Talkolonner er noget, der er god brug for. I hvert fald har vi fået flere henvendelser om emnet efter sidste udgave af VI & VIC, hvor vi angav en subrutine, der kunne klare sagen.

En svaghed i den viste rutine var hastigheden, idet afvigelser fra den ønskede linielængde blev kompenseret med én position ad gangen - hvorefter der igen blev testet for korrekt linielængde.

Bent R. Husted i Esbjerg skriver således til os:

Efter at have studeret jeres subrutine i VI & VIC nr. 2/83 vedr. kr. og øre præsentation - en facilitet, som jeg ofte har savnet på min VIC, synes jeg, at jeg ville tillade mig at bidrage med en tilsvarende rutine, der har et par fordele frem for det præsenterede program, idet jeg straks blev slået af sub'ens langsomhed. Det er ikke lykkedes mig, efter mange forsøg, at få følgende programstump til at svigte i sin tilsigtede funktion.

Linie 904 checker om 'pr' er en ren decimalbrøk og hægter i givet fald et '0' foran tallet.

Såfremt det tal, der skal printes, er negativt, sørger linie 905 for at stable et '-' foran, dette blev jo fjernet i linie 903. Hvorfor så linie 902? Jo, VIC omsætter decimalbrøker, der er mindre end '.01 til eksponentialnotation (f.eks. bliver tallet 0.001 udskrevet som 1E-03), dette vil ske, såfremt tallet, der skal printes, er NUL p.g.a. metoden i linie 903, - derfor.

```
900 tr$=left$(tr$+".....................",11):goto902
901 tr$=left$(tr$5"                     ",11)
902 if pr=0 then pr$="0.00":goto906
903 pr$ = str$(int(abs(pr)*100+.5)/100+.001):pr$ = mid$
    (pr$,2,len(pr$)-2)
904 if abs(pr)<1 then pr$ = "0"+pr$
905 if pr<0 then pr$ = "-"+pr$
906 print left$(tr$,11-len(pr$));pr$ : return
```



Fra Ole Andersen i Ålborg har vi modtaget et andet forslag til løsning af opgaven:

Lige et par ord for at fortælle jer, at jeg har fundet ud af at stille tal op under hinanden i kolonner, så det ser ordentlig ud.

Subrutinen er en videreudvikling af mit eget hidtil anvendte »kolonniseringsprogram« - der ikke indeholdt tekstningen (tr$) - endvidere er det skrevet således, at det blot ved ændring af linienumre direkte kan indgå i programmer der allerede er skrevet med VI & VIC's subrutine. I det følgende vil jeg kort redegøre for programmets funktion.

Indsætning af prikker eller spaces mellem tekst og tal foretages valgfrit ved at gå ind i linie 900 eller 901 (left$ for at forhindre evt. overskridelse af tilladelig strenglængde (!)).

Problemet med efterfølgende nuller klares ved at »snyde« en ekstra decimal (.001) på tallet efter afrunding i linie 903, denne fupdecimal og det indledende fortegnsfelt fjernes i strengen (pr$) i sidste sætning i linien. For ikke at skulle lave et større nummer ud af det, dannes strengen 'pr$' ud fra ABSolutværdien (altid positiv) af tallet 'pr'.

X = det tal, der skal udskrives, og det bliver lavet om til X$ og bliver udskrevet som sådan. Jeg håber, at der er andre, der kan få glæde af dette.

*Ole Andersen*

```
62000 A$=RIGHT$("          "+STR$(INT(X)),10)
62010 B$=LEFT$(MID$(STR$(INT((X-INT(X))*100+.005))+"00",2),2)
62020 C$="?":X$=A$+C$+B$:RETURN
```

# Kommunikationen til printer ikke i orden

Fra en læser i Norge, Per Egil Tellefsen i Hønefoss, har vi fået præsenteret et problem med en printer, som ikke vil printe. Det drejer sig om en »Digital LA34«, som brugeren har sat i forbindelse med VIC ved hjælp af et RS 232-interface. Han får ingen forbindelse og spørger derfor, om det kan tænkes, der er forskel på ind- og udgangene.

Nu er kommunikation med ydre enheder en mere kompliceret sag end blot at sætte et stik rigtigt i. Vi tror ikke, det er ben-figurationen, der er noget galt med, men man kan forsøge sig med at ombytte ledningerne til ben 2 og ben 3.

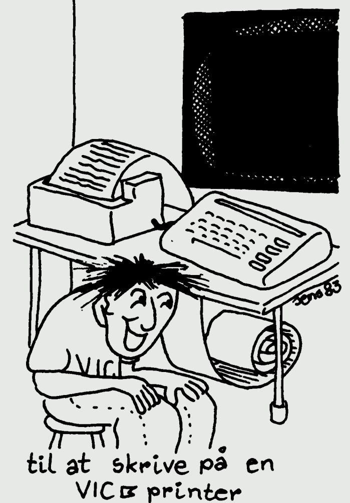
Vær opmærksom på, at RS 232 benytter sig af apparat nummer to, så der skal altså ikke åbnes til nummer 4, som når der bruges en printer, der kan tilsluttes direkte til VIC.

Der skal i stedet åbnes til apparat 2, f.eks. OPEN 1,2,1.

Men efter denne åbning skal der skrives en CHR$-værdi, der beskriver kommunikationen. Det er vigtigt at sende til printeren med samme hastighed, som den forventer at modtage. Der kan sendes med hastigheder fra 50 til 2400 baud. 1200 er en sædvanlig indstilling, men det skal altså lige tjekkes efter.

Desuden må ordlængden og antallet af STOP-bits defineres. dette findes der flere informationer om i bogen VIC Revealed (side 204-215).

# Sæt skærmen ud af drift, når C-64 skal bruge VIC printer og diskette-drev




Commodore 64 er en væsentligt kraftigere sag end VIC-20, og det kan give visse vanskeligheder, når udstyr skal benyttes til begge maskiner.

Det gælder således diskettestationen og printeren, men det har vist sig at være muligt at få Commodore 64 til at køre med VIC-udstyret.

Problemet består i hastighederne, hvormed data overføres, og løsningen er, at skærmen sættes ud af drift, så Commodore 64 ikke behøver bruge energi på at generere skærmbilledet.
Det skulle være tilstrækkeligt til at både printer og diskettestationen kan køre uden problemer.

Skærmen sættes ud af drift med ordren POKE 53265,11 - og når man skal tilbage til normal skærm, er ordren 53265,27.
Skulle der opstå problemer med at komme tilbage til den normale skærm, skulle det kunne klares ved at trykke på RUN/STOP og RESTORE på samme tid.

# Fejlene i VIC 20 brugervejledning

Under sætningen af VIC-20 brugervejledning er der opstået nogle fejl.
Her bringer vi hele listen af rettelser.
Fejl er næsten umulige at undgå, og der er også konstateret skønhedsfejl i Commodore 64-vejledningen. Når en egentlig fejlliste foreligger, vil vi også bringe den videre, men her er i første omgang VIC-20-smutterne:

Side 26 På sidens øverste halvdel: De to grafiske afbildninger af tasterne: RVS ON og RVS OFF, skal byttes om.

Side 28 Programeksempel: 8GOTO50. Skal være: 8GOTO5.

Side 29 I 4. tekstlinie: i hele 1å. Skal være: i hele 16.

Side 36 Tekst i kasserne: Memory location. Betyder: Hukommelses plads.

Side 42 I 2. linie. I mellemrummet mellem 'RUN' og 'igen', skal der være et 'RETURN'-mærke. (Som i linie 1).

Side 53 Sidste linie på siden: og tryk på . I mellemrummet skal der være et 'RETURN'-mærke.

Side 68 Linie 130. Skal være: FOR N = 1 TO 2000 : NEXT N

Side 76 Øverst: 7. tone hedder BI, ikke TI.

Side 77 Variablen i linierne 120, 130 og 150 er TØ (t-nul), og ikke TO, der er reserveret variabel, og derfor ulovlig.

Side 82 Linie 30 i eksemplet: x skal ændres til *, (gangetegn).

Side 83 Mellem 'HVAD' og 'ER' i linie 1, skal der være et 'SPACE'-mærke.

Side 103 X = RND(1)10 + 10, skal være: X = RND (1)*10 + 10
X = INT(RND(1)10 + 10), skal være:
X = INT(RND(1)*10 + 10)

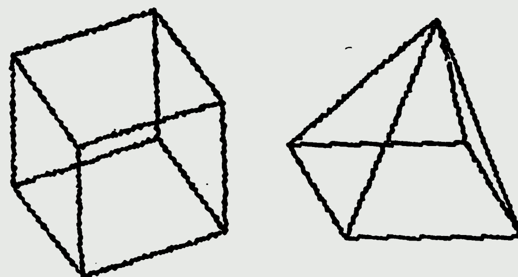
Side 104 X = INT(RND(1)11 + 10), skal være:
X = INT(RND(1)*11 + 10)
X = INT(RND(1)a) + b, skal være:
X = INT(RND)(1)*a) + b

Side 107 IEEE 488 INTERFACE. Findes ikke.

Side 111 Linie 70, skal være: PRINT = 1,A$
Linie 80, skal være: IF A$ <> "STOP" THEN 60

Side 153 Under indtastningen af dette program, kan der opstå problemer i linie 135. Denne linie er nøjagtig 88 tegn lang - hvis man indtaster uden mellemrum - hvilket gør den meget svær at indtaste. Gør følgende: Forkort alle PEEK-instruktioner til: P shiftet E, alle POKE til P shiftet O, osv. (Se side 133). Sørg for at markøren står indenfor de fire første skærmlinier, før der trykkes på RETURN.

# 200 sider med grafik for VIC 20 med super-expander



Det er en verden af nye muligheder, der åbner sig for den VIC-bruger, der har anskaffet en super-expander, ikke mindst med hensyn til grafikken.

Super-expanderen giver virkelig mange muligheder for at lege med farver og grafiske effekter på skærmen - men hvordan kommen man i gang, når man har leget færdig med brugervejledningens eksempler? Een af måderne er at kigge i Nick Hampshires bog »VIC Graphics«. Den er på engelsk, men hvis man tør kaste sig ud i læsning af dette i data-verdenen dominerende sprog, er her næsten 200 siders inspiration til super-expander-brugeren.

I bogen gennemgås de specielle BASIC-kommandoer, som super-expanderen giver mulighed for, f.eks. CIRCLE og PAINT, og der er masser af program-eksempler, der illustrerer, hvordan skærmbilledet bygges op.

Bogen er inddelt i kapitler, der hver for sig behandler sit aspekt af brugen af grafik, og det program-eksempel, vi har hentet fra bogen stammer fra kapitlet om tre-dimensionelle figurer.

VIC Graphics koster i Danmark 195 kr., og den kan fås hos VIC forhandlerne.

```
1 REM 3D DRAWING 1
2 REM ************************
3 REM
10 REM A THREE DIMENSIONAL SHAPE IS DRAWN BY THIS PROGRAM
20 REM THE ROTATION POSITION AND SCALE OF THE OBJECT
30 REM CAN BE CHANGED TO GIVE DIFFERENT VIEWING ANGLES.
35 REM
40 REM SET COLOURS
50 GRAPHIC 2
60 COLOR 3,3,0,10
65 REM
70 REM DRAW BORDER AROUND SCREEN
75 REM
80 GOSUB 900
85 REM
90 REM SET UP CONSTANTS, VARIABLES, AND ARRAYS
95 REM
100 DIM A(4;4)
110 DIM B(4,4)
120 SX=.3
130 SY=.3
140 SZ=.3
150 TX=1
160 TY=1
170 TZ=1
180 RX=40*π/180
190 RY=20*π/180
200 RZ=50*π/180
400 REM MAIN PROGRAM LOOP
410 GOSUB 1000
420 GOSUB 5000
430 GOSUB 3000
440 GOSUB 4000
450 GOSUB 2000
500 GET A$:IF A$="" THEN 500
510 COLOR 1,3,6,0
520 GRAPHIC 0
530 END
900 REM BORDER DRAWING SUBROUTINE
905 REM
910 POINT 3,0,0
920 DRAW 3 TO 0,1023
930 DRAW 3 TO 1023,1023
940 DRAW 3 TO 1023,0
950 DRAW 3 TO 0,0
960 RETURN
995 REM
1000 REM INITIALISE SHAPE
1005 REM
1010 NP=8
1020 NE=12
1030 REM
1040 DIM S(3,NP)
1050 DIM E(NE,2)
1060 DIM M(3,NP)
1100 REM
1110 FOR N=1 TO NP
1120 READ S(1,N),S(2,N),S(3,N)
1130 NEXT N
1140 FOR K=1 TO NE
1150 READ E(K,1),E(K,2)
1170 NEXT K
1195 REM
1200 REM X,Y,Z POINT COORDINATES
1210 DATA 0,0,200,200,0,200,200,0,0,0,0;0
1220 DATA 0,200,200,200,200,200,200,200,0,0,200,0
1295 REM
1300 REM CONNECTION DATA
1305 REM
1310 DATA 1,2,2,3,3,4,4,1
1320 DATA 5,1,2,6,4,8,7,3
1330 DATA 6,5,5,8,8,7,7,6
1900 RETURN
1995 REM
2000 REM DRAW SHAPE
2005 REM
2020 FOR K=1 TO NE
2030 V1=E(K,1)
2040 V2=E(K,2)
2045 IF V1=0 THEN 2240
2050 XB=M(1,V1)
2060 YB=M(2,V1)
2070 XE=M(1,V2)
2080 YE=M(2,V2)
2090 DS=1
2100 P=XE-XB
2110 Q=YE-YB
2120 R=SQR(P*P+Q*Q)
2130 LX=P/R
2140 LY=Q/R
2150 FOR I=0 TO R STEP DS
2160 X=6*0.7*(XB+I*LX)
2170 Y=1023-6*(YB+I*LY)
2180 IF X<0 OR Y<0 THEN 2230
2190 IF X>1023 OR Y>1023 THEN 2230
2220 POINT 3,X,Y
2230 NEXT I
2240 NEXT K
2900 RETURN
2995 REM
3000 REM SET TRANSFORMATION MATRIX
3005 REM
3010 A(1,1)=COS(RY)*COS(RZ)
3020 A(1,2)=COS(RY)*SIN(RZ)
3030 A(1,3)=-SIN(RY)
3040 A(1,4)=0
3050 A(2,1)=COS(RX)*(-SIN(RZ))+SIN(RX)*SIN(RY)*COS(RZ)
3060 A(2,2)=COS(RX)*COS(RZ)+SIN(RX)*SIN(RY)*SIN(RZ)
3070 A(2,3)=SIN(RX)*COS(RY)
3080 A(2,4)=0
3090 A(3,1)=(-SIN(RX))*(-SIN(RZ))+COS(RX)*SIN(RY)*COS(RZ)
3100 A(3,2)=-SIN(RX)*COS(RZ)+COS(RZ)*SIN(RY)*SIN(RZ)
3110 A(3,3)=COS(RX)*COS(RY)
3120 A(3,4)=0
```

*Fortsættes side 10*

*Fortsat fra side 9*

```
3130 A(4,1)=0
3140 A(4,2)=0
3150 A(4,3)=0
3160 A(4,4)=1
3195 REM
3200 REM SET UP SCALING AND TRANSLATION MATRIX
3205 REM
3210 B(1,1)=SX*A(1,1)
3220 B(1,2)=SX*A(1,2)
3230 B(1,3)=SX*A(1,3)
3240 REM
3250 B(2,1)=SY*A(2,1)
3260 B(2,2)=SY*A(2,2)
3270 B(2,3)=SY*A(2,3)
3280 REM
3290 B(3,1)=SZ*A(3,1)
3300 B(3,2)=SZ*A(3,2)
3310 B(3,3)=SZ*A(3,3)
3320 REM
3330 B(4,1)=TX
3340 B(4,2)=TY
3350 B(4,3)=TZ
3900 RETURN
3995 REM
4000 REM PERFORM TRANSLATION
4005 REM
4010 FOR Q=1 TO NP
4015 REM
4020 XT=S(1,Q)-XC
4030 YT=S(2,Q)-YC
4040 ZT=S(3,Q)-ZC
4045 REM
4050 M(1,Q)=XC+(XT*B(1,1)+YT*B(2,1)+ZT*B(3,1)+B(4,1))
4060 M(2,Q)=YC+(XT*B(1,2)+YT*B(2,2)+ZT*B(3,2)+B(4,2))
4070 M(3,Q)=ZC+(XT*B(1,3)+YT*B(2,3)+ZT*B(3,3)+B(4,3))
4080 NEXT Q
4900 RETURN
4995 REM
5000 REM FIND CENTROID
5005 REM
5010 P=0:Q=0:R=0
5020 FOR I=1 TO NP
5030 P=P+S(1,I)
5040 Q=Q+S(2,I)
5050 R=R+S(3,I)
5060 NEXT I
5070 XC=P/NP
5080 YC=Q/NP
5090 ZC=R/NP
5900 RETURN

READY.
```

# Masser af programmer til Commodore 64 og til VIC 20

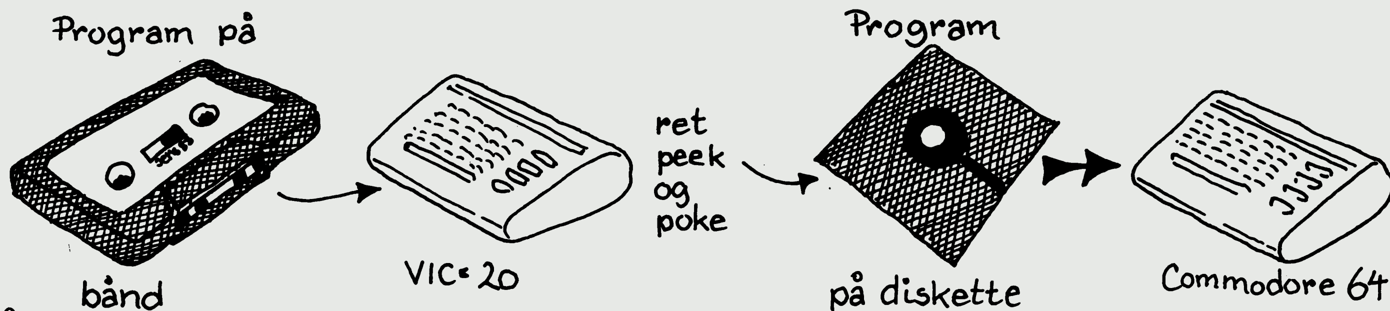
Der er stadig mange Commodore 64-brugere, der søger råd om, hvordan VIC-programmer læses ind i Commodore 64.
Som omtalt i sidste nummer af VI & VIC skal programmerne modificeres, hvis de indeholder PEEK eller POKE kommandoer, ligesom der skal tages højde for at Commodore 64 har 40 karakterer på hver skærm-linie, hvor VIC 20 kun har 22. Endelig skal eventuelle maskinkodeprogrammer skrives om. Selve indlæsningen fra kassettebånd kan ikke foregår direkte, her må man gøre omvejen over en diskettestation.

Et VIC-program på kassette skal altså først indlæses i en VIC 20, hvorfra det lagres på diskette. Fra denne diskette kan det indlæses i Commodore 64, hvorfra det igen kan lagres på kassette, hvis det er den opbevaringsform, man har valgt.

Men til glæde for de mange, som med spænding har ventet på færdige programmer til Commodore 64 kan det oplyses, at der er godt nyt på vej.

Ikke alene er der mange forskellige spil undervejs, men der er allerede udgivet båndprogrammer med titlerne »Matematisk kryds og bolle«, »Gi' og tag«, »Matematikmodul« (der udregner ligninger) og et »kurvetegningsprogram«. Disse programmer koster 125 kr. pr. stk. hos VIC forhandlerne.

På diskette fås nu fem nye programmer, der findes både til Commodore 64 og til VIC 20.
Det er et bogføringsprogram, et lagerprogram, et multikartotek, hvor der kan søges med to kriterier ad gangen, et plakat-udskriftprogram med kæmpebogstaver, og endelig et faktureringsprogram.



# Computeren må gerne blive varm

Tvivlere har spurgt, om det virkelig kan være rigtigt, som vi tidligere har skrevet, at man kan lade være med at slukke for sin computer. Den bliver jo varm, og kan den nu tåle det.
Det samme sprøgsmål er kommet fra en VIC-skakspiller, som i brugervejledningen til SARGON II Chess har hæftet sig ved oplysningen om, at den gennemsnitlige træktid i højeste sværhedsgrad kan blive fire timer. Kan VIC virkelig holde til at være tændt så længe? spørger Jesper Sørensen i Allerød.
Vi kan føje til, at det også står i brugervejledningen til SARGON II Chess, at de enkelte træk kan variere fra en trediedel til tre gange den gennemsnitlige træktid, så man kan altså komme ud for, at VIC skal gruble over et enkelt skak-træk i 12 timer(!) - og vi kan straks trøste den tålmodige skak-spiller med, at det kan computeren sagtens tåle. Det er naturligvis en god skik at slukke for elektriske apparater, når de ikke bruges, men der er ingen grund til at frygte, at VIC 20 eller Commodore 64 skal »brænde sammen«. Hos Commodore har man eksemplarer af VIC 20, som har været tændt i flere måneder, og det giver ikke anledning til vanskeligheder.
Tænk også på, at computeren kan anvendes i kontrolsystemer, der ganske enkelt forudsætter, at den er i drift konstant.

## Varmen

Det er varmen fra selve computeren og fra den lille transformator, der giver anledning til bekymring. Og det er da også rigtigt, at begge dele udvikler varme, men det er en helt normal foreteelse. Varmeudviklingen bliver kun et problem, hvis computeren ikke kan slippe af med sin overskydende energi.
Anbring derfor altid computeren på en plan bordplade, så der kan komme luft ind under den - der er luftåbninger i bunden af kabinettet - og de må naturligvis ikke lukkes.
Hindrer man computeren i at slippe af med sin varme, vil det være et spørgsmål om tid, hvornår kabinettet smelter.
Når computeren står på et bord og der er normal ventilation omkring den er der absolut intet faretruende ved varmen.
Samme forholdsregler bør tages for alle andre apparater, der afgiver varme - selv en almindelig skrivebordslampe har sine ventialtionsriller - men i computersammenhæng er det måske specielt



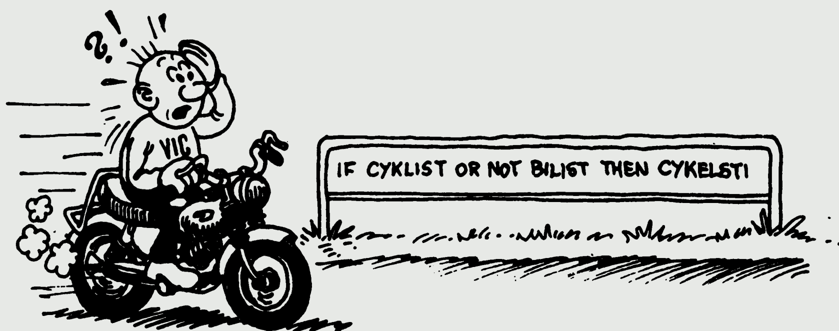

diskette-drevet, der stiller krav om ventilation. Det er så fristende at give diskettestationen sin lille plads i bogreolen, så man bare kan putte disketterne ind i fronten, men her må man være sikker på, at luften kan cirkulere frit omkring apparatet.

# Vic 20 og Commodore 64 på Herning-messen

Der vil blive lagt mærke til de små Commodrecomputere, når HI-messen åbner i september i Herning. Commodore Data A/S har fået sin egen boble-hal på 300 kvadratmeter, og det bliver først og fremmest VIC 20 og Commodore 64, der vil blive præsenteret på denne udstilling. For brugere, der allerede har en computer, bliver der også nok at se på, hvis man har mulighed for at komme til Herning. På udstillingen vil der blive præsenteret udstyr til computerne, og mange af de nye spil, som kommer på markedet til efteråret er klar til demonstration på udstillingen.

# VIC revealed

VIC Revealed er for mange VIC 20-brugere blevet en lille bibel, der gav svar på mange af de ting, der ellers kunne mystificere programmørens tilværelse. Efter at have været udsolgt er bogen (der er på engelsk) kommet i en ny udgave. Prisen hos VIC forhandlerne er 195 kr.




# Ny dansk bog om programmering med Commodore BASIC er skrevet direkte til VIC 20

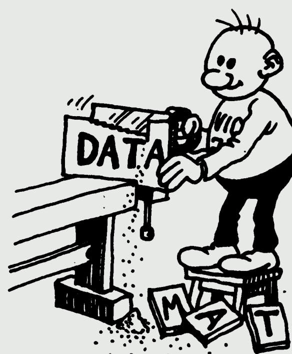
Brugervejledningerne til de små computere fra Commodore, VIC 20 og Commodore 64, er så instruktive, at man faktisk kan lære BASIC-programmering blot ved at læse dem - og hele tiden have dem ved hånden, når man arbejder.
Ikke desto mindre har efterspørgslen efter mere programmeringslekture været meget stor, men nu er den kommet - på dansk.
Det er Erwin Neutzsky-Wulff, som på BORGEN's forlag har fået udgivet bogen »PROGRAMMERING med Commodore BASIC«. Det er 220 siders spændende VIC 20-læsning for de 148 kr., bogen koster.
Bogen er skrevet, så den kan læses helt uden forudsætninger - og når man er kommet igennem den har man et virkeligt godt grundlag for at gå videre til den mere avancerede data-litteratur og til at forstå datatidskrifternes undertiden meget indforståede artikler.
»PROGRAMMERING med COMMODORE BASIC« er skrevet som en lærebog med 40 lektioner. Har man lyst, kan man ganske enkelt starte ved lektion 1 og arbejde sig igennem bogen - men den er også udmærket som opslagsværk - bl.a. fordi der er et godt stikordsregister ligesom indholdsfortegnelsen er så overskuelig, at man let finder frem til den lektion, man har brug for at »repetere«.

### Fra bar bund

Første del har 10 lektioner for den VIC-bruger, der er helt på bar bund - og også en god introduktion til den læser, der endnu ikke har en computer. Her får man noget at vide om, hvad BASIC egentlig er for noget, hvordan variabler fungerer og hvordan programmer afvikles. Logiske operationer og relationer behandles også i denne del.
I anden del er det data og sæt af variabler, der får tildelt den største opmærksomhed, tredie del er helliget billede og lyd, mens fjerde del drejer sig om karakterer og grafik.

### Mange eksempler

En positiv ting ved bogens




gennemgang af programmeringsmulighederne er, at der konsekvent er anført eksempler på, hvordan de forskellige funktioner kan bringes i anvendelse. Små lette programmer, som er lette at taste ind, så man hurtigt kan få syn for sagn: Det virker faktisk som beskrevet.
Rart er det også, at der er blevet plads til en oversigt over nogle af de system-variabler, som man ellers kan have svært ved at finde frem til.
»PROGRAMMERING med COMMODORE BASIC« er illustreret af Jesper Deleuran.

## Du kan købe bogen gennem VI & VIC

Bogen om programmering kan du bestille hos VI & VIC. Den koster 148 kr. Brug bestillingskuponen her i bladet eller skriv til: VI & VIC, Kornager 29, 7100 Vejle.

# Andre Joy-sticks kan bruges til VIC

*Vi har modtaget:*

Jeg er en spillefanatiker på 14 år. Dog spiller jeg kun med »goe'e gamle« VIC 20. Jeg har derfor anskaffet mig et Commodore Joy-stick. Dette Joy-stick finder jeg, og sikkert mange andre, dårligt. Laver Commodore virkelig ingen bedre joy-stick? Hvis ikke - Kan joy-stick fra andre computere bruges, f.eks. et fra Texas Instrument? (Måske med en lille ombygning!?).
Jeg vil sætte stor pris på jeres svar.

Venlig hilsen den spilleglade.
Jesper Sørensen
Løvsangervang 8
3450 Allerød

Om en joy-stick er god nok eller ej er op til den enkelte at bedømme, men hvis man ønsker at bruge andre fabrikater, skulle det normalt ikke give vanskeligheder. F.eks. kan det oplyses, at Commodores joy-stick er Atari-kompatibelt - eller på dansk: Atari's joy-stick passer til Commodore-computerne.

# Mikroprocessorens status register og brugen af »flag«

Processorens status register indtager en meget vigtig plads i 6502's interne opbygning. Det er et otte bit programmerbart register, og dets funktion ligger midt mellem kontrol- og registerdelen. Det er det eneste register der har nogen egentlig indflydelse på kontrol-logikken. Syv af de otte bit bruges, og hvert bit, eller flag, har en nøje bestemt funktion.

### Tre slags flag

Flag kan deles op i tre kategorier, de der kan kontrolleres af programmøren alene, de der kun kan kontrolleres af processoren. Der er i 6502 kun eet flag der falder ind under den første kategori, decimal flaget, eller D-flaget. Det er bit tre i status registreret. Dette flag bestemmer hvorvidt processoren udfører binære eller decimale beregninger. Det kan sættes med en sed instruktion, hvorefter al beregning foregår decimalt. Bittet kan slettes med cld, hvorefter al beregning foregår binært.

Der falder tre flag ind under den anden kategori: carry, overflow og interrupt disable. Carry eller C-flaget er bit nul, og ændres enten af visse beregninger, eller af programmøren. Carry bruges også ofte som det »niende« bit under beregninger, eller under shift og rotate instruktioner. Instruktionen der benyttes for at sætte carry er sec, og det kan slettes med clc.

Overflow eller V-flaget er bit seks, og benyttes under beregninger hvor det bruges til at indikere, at beregningen krævede et resultat der var større end de syv byte der er plads til, (ottende bit er fortegn.). V-flaget har samme betydning som carry flaget, men indikerer også at fortegnet må genoprettes, da overflowet i bit syv vil have ødelagt det originale fortegn. Programmøren kan kun slette V-flaget med clv instruktionen.

### Interrupt

Interrupt disable, I-flaget, styrer virkemåden af processorens interrupt input, og er placeret i bit to. Interupt er en meget vigtig del af en Commodore computers opbygning, og hver gang der forekommer et interrupt, sætter processoren I-flaget. Dette forhindrer processoren i at modtage flere interrupter på irq linien, indtil programmet har behandlet den først ankomne interrupt korrekt, og afsluttet med at slette interrupt disable flaget. I-flaget kan også sættes af programmøren, med instruktionen sei, f.eks. for at forhindre afbrydelse i en tidskritisk programløkke. I slutningen af en sådan rutine kan interrupten igen sættes i funktion med instruktionen cli.

De sidste tre flag: zero, negative og break, kontrolleres alene af processoren. Zero og negative flagene sættes eller resættes af næsten enhver processor operation. Z-flaget sættes af processoren på ethvert tidspunkt, hvor resultatet af en operation er nul, som hvis to tal med samme værdi trækkes fra hinanden. N-flaget sættes af processoren altid lig med det syvende bit af resultatet af en operation. Et af de vigtigste formål med dets brug, er i forbindelse med beregninger, idet N-flaget sættes hvis resultatet er negativt. B-flaget sættes af processoren medens den er igang med at færdiggøre en interrupt rutine. Z-flaget er bit et, N-flaget er bit syv og B-flaget er bit fire.

### Som i BASIC

De syv status flag har hver især en betydning for programmøren på forskellige stadier i en programudvikling. Selv om carry og overflow flagene bruges meget i beregninger, er den største brug af dem i de betingede hop-instruktioner. Dette giver programmøren mulighed for, at indlægge programforgreninger, der er afhængige af flag, for på denne måde, at lade resultatet af f.eks. en beregning være afgørende for programmets videre gennemførsel. Dette er i princippet helt det samme som man i basic kan opnå med instruktioner som..if..then.goto. Der er et stort udvalg af instruktioner af denne art. Enhver, der skriver et program, må være helt klar over hvilken stand de forskellige flag forventes at være i, igennem hele programmet. At undlade dette er en af de allermest almindelige grunde til, at maskinkode programmer enten ikke vil virke, eller giver de forkerte resultater. Et eksempel ville f.eks. være at undlade at slette carry flaget før en addition. Det kunne måske være blevet sat under forudgående instruktioner, og ville uværgeligt give forkerte resultater.

# Forbedring af ØKO-SYS programmet, så det kan bruges med printer

Verner Madsen, Tårnvej 265, 5. i Rødovre har skrevet en udvidelse til VI & VIC's ØKO-SYS program, der sætter det i stand til at give output til en printer.

Han sender en programlistning af de nye og ændrede programlinier samt følgende forklaring:

I linierne 1151, 1271 og 1821 er værdien »> 56« ændret til »>57«, som er ASCII-koden for tallet 9. Dette bevirker, at du ved tryk på 9, ifølge den understregede tilføjelse i linie 1160, kommer ned i subrutine 61000. Dette gælder dog ikke, når du ved spørgsmålet: »Hvilken konto«?, trykker på 9, ej heller ved indtastning af 9-taller i beløbene.

Subrutinen i linie 61000-61140 er den samme som i brugervejledningen til printeren, dog med få ændringer.

```
1151 IFASC(S$)<49ORASC(S$)>57THEN1150
1160 ONSGOSUB50000,3000,1200,1500,2100,1700,1800,1900,61000
1271 IFASC(S$)<49ORASC(S$)>57THEN1270
1821 IFASC(S$)<49ORASC(S$)>57THEN1820
61000 REM SKÆRMKOPI
61010 G1$=CHR$(145)
61020 OPEN4,4:PRINT#4:G1=4074
61030 FORG0=0TO22:G0$=G1$:G1=G1+22
61040 FORG2=G1TOG2+21:G3=PEEK(G2)
61050 IFG3>128THENG3=G3-128:G4=1:G0$=G0$+CHR$(18)
61060 IF(G3>0)*(G3<32)THENG3=G3+64:GOTO61100
61070 IF(G3>31)*(G3<64)THEN61100
61080 IF(G3>63)*(G3<96)THENG3=G3+128:GOTO61100
61090 IF(G3>95)*(G3<128)THENG3=G3+64:GOTO61100
61100 G0$=G0$+CHR$(G3)
61110 IFG4=1THENG0$=G0$+CHR$(146):G4=0
61120 NEXTG2:PRINT#4,CHR$(15)G0$CHR$(15):NEXTG0
61140 RETURN
```

# Funktionstasterne gør livet lettere: Du skal bare trykke på én knap

Ville det ikke være dejligt, hvis man f.eks. kunne starte et program, blot ved tryk på én taste. Eller, under programmeringen blot trykke på den første funktionstaste og få programmet LISTet på skærmen?

Her viser vi, hvordan det gøres:

Selvom programmet, der gør funktionstasterne programmerbare, er i maskinkode, behøver du ingen kendskab til maskinkode, for at bruge det.

Når programmet er indtastet, skal du sørge for at sammenligne programmet med listningens DATA-sætninger, da en enkelt fejl kan få din VIC til at gå i »baglås«. RUN først programmet, når du har SAVET det. Derefter fremkommer følgende: "F1 = ?". Indtast den BASIC-kommando, eller instruktion, du ønsker funktionstaste nr. 1 skal have. For hver RETURN, der skal inkluderes, trykker du på '←' pil til vesntre. Når du er sikker på, at tasten er defineret korrekt, trykker du på RETURN.

Programmet vil så spørge efter de resterende funktionstasters definitioner. Når du har programmeret den 8. taste, vil computeren skrive READY. Nu kan du blot trykke på den ønskede taste, og de tegn tasten har fået tildelt, vil blive printet.

## Således virker programmet

Hvis din VIC går i baglås, eller du ikke får READY til sidst, må du slukke for computeren, tænde igen, og genindlæse programmet fra kassetten. Sammenlign igen programmet med listningen fra begyndelsen til linie 65, specielt DATA linierne.

Hvis READY kommer på skærmen, efter de 8 taster er defineret, og VIC ikke skriver de tegn, der blev defineret, ved tryk på tasterne, skal du undersøge programmet fra 70 til 95.

Hvis det stadig ikke virker, undersøges linie 100-115.

## Hvad nu, hvis det ikke virker

BASIC programmet POKER to maskinkodeprogrammer ind i din VIC. Det ene går ned i kassettebufferen, det andet i den øverste del af RAM'en.

Programmet i kassettebufferen spørger efter difinitionerne på de otte taster.

Når du trykker på RETURN, gemmer programmet din indtastning i den øverste del af RAM'en. Når alle otte taster er programmeret, vil det først program få computeren til at gå til det andet program i toppen af RAM'en, hver 60. sekund. Herefter bruges det originale program ikke, og det slettes automatisk ved næste brug af kassettebåndoptageren. Herved spares 147 bytes af VIC's hukommelse.

Det andet program undersøger konstant, om der er trykket på en funktionstaste. Hvis der er, vil programmet udskrive de tegn, der er defineret for tasten.

## Sådan sparer du plads

Jo længere hver kommando til en funktionstaste er, jo mere hukommelse bruges der. Hvis kommandoerne er korte, vil de fylde ca. 200 bytes. Programmet er maksimalt istand til at bearbejde 800 bytes. For at spare på pladsen, kan kommandoerne indtastes i forkortet form, som vist i VIC-20 brugervejledning side 133 og 134.

De programmerbare taster kan virkelig spare programmøren for meget indtastning. VIC er udstyret med 8 funktionstaster, som du nu kan bruge til lige hvad du har lyst til.

Her er et eksempel:

RUN
F1 = ? LIST
F2 = ? POKE
F3 = ? RUN
F4 = ? PEEK (
F5 = ? GOTO
F6 = ? GOSUB
F7 = ? PRINT PEEK (7680)
F8 = ? LOAD LIST

Udover denne måde at bruge funktionstasterne på, findes der også en mere simpel, i ren BASIC. Den er vist i VIC-20 brugervejledning side 147.

```
5 F=0:C=PEEK(55)-120:IFC<0THENC=C+256:F=-1
10 D=PEEK(56)+F:POKE55,C:POKE56,D:CLR
15 S=828:I=146:GOSUB100
20 DATA32,198,3,165,55,133,251,133,253,165,56,133,252,133,254,169,49,133,0,169
25 DATA133,133,1,169,13,32,210,255,169,70,32,210,255,165,0,32,210,255,169,61
30 DATA32,210,255,169,63,32,210,255,169,32,32,210,255,32,207,255,72,160,0,165
35 DATA1,145,55,104,32,198,3,201,13,240,14,201,95,208,2,169,13,145,55,32
40 DATA207,255,76,124,3,230,0,165,0,41,1,208,10,24,165,1,105,4,133,1
45 DATA76,170,3,56,165,1,233,3,133,1,165,0,201,57,144,163,120,169,L0,141
50 DATA20,3,169,H0,141,21,3,88,169,0,133,0,32,68,198,76,116,196,166,55
55 DATA208,2,198,56,198,55,96
60 S=PEEK(55)+256*PEEK(56):I=119:GOSUB100
65 SYS 828
70 DATA165,0,240,59,160,0,177,251,32,L99,H0,176,12,165;55,197,251,208,21,165
75 DATA56,197,252,208,15,169,0,133,0,165,253,133,251,165,254,133,252,76,191,234
80 DATA166,198,177,251,157,119,2,230,198,32,L111,H0,165,198,201,11,144,204,230,0
85 DATA76,191,234,165,215,32,L99,H0,176,3,76,191,234,165,8,41,1,208,247,160
90 DATA0,177,251,197,215,208,6,32,L111,H0,76,L6,H0,32,L111,H0,76,L81,H0,201
95 DATA133,144,6,201,141,176,2,56,96,24,96,166,251,208,2,198,252,198,251,96
100 F=0:FORD=STOS+I:READA$:IFASC(A$)<58THENA=VAL(A$):GOTO115
105 IFASC(A$)=76THENA=VAL(RIGHT$(A$,LEN(A$)-1))+PEEK(55):IFA>255THENA=A-256:F=1
110 IFASC(A$)=72THENA=VAL(RIGHT$(A$,LEN(A$)-1))+PEEK(56)+F:F=0
115 POKED,A:NEXT:RETURN
```

# PLAKAT PRINTER

```
100 REM    **************************************************
110 REM    *                                                *
120 REM    * PLAKAT PRINTER                        MAJ 1983 *
130 REM    * PROGRAMMØR:   CHRISTIAN DIRKSEN                *
140 REM    *                                                *
150 REM    *------------------------------------------------*
160 REM    * FORMAAL :                                      *
170 REM    *        UDSKRIFT AF FORSTØRREDE TEGN (8 GANGE 8) *
180 REM    *        PAA PRINTER AF SEIKOSHA GP-100VC TYPEN.  *
190 REM    *        TILPASSES LET TIL ANDRE PRINTERE ELLER   *
200 REM    *        TIL SKAERMEN.                            *
210 REM    *------------------------------------------------*
220 REM    * VARIABLER:                                     *
230 REM    *         B(I)  TABEL OVER POTENSER AF 8. BRUGES  *
240 REM    *               TIL BIT-SAMMENLIGNING.           *
250 REM    *         C$    DET TEGN, PLAKATKARAKTERERNE OP-  *
260 REM    *               BYGGES AF.                       *
270 REM    *         S$    TEKSTSTRENGEN, DER FORSTØRRES OP. *
280 REM    *         T     LAENGDEN AF TEKSTSTRENGEN S$.     *
290 REM    *         S     HVIS SAND : STANDARD KARAKTERER,  *
300 REM    *               HVIS FALSK: PROGRAMMEREDE TEGN.   *
310 REM    *         CG    KARAKTERGENERATORENS PLACERING I  *
320 REM    *               RAM/ROM.                         *
330 REM    *         J,P,I TAELLERE I 'FOR-LOOP'.            *
340 REM    *         A     ASCII-VAERDI AF HVERT TEGN I S$.  *
350 REM    *         K     EN BYTE I DEN KARAKTER, DER BE-   *
360 REM    *               HANDLES 'NU'.                    *
370 REM    *         A$    HVIS 'J' : FORTSAET,              *
380 REM    *               HVIS 'N' : FAERDIG.               *
390 REM    *                                                *
400 REM    **************************************************
410 :
420 OPEN 1,4:          REM * TIL SKAERM: 'OPEN 1,3' *
430 FOR I=0 TO 7
440 :  B(I)=2↑I
450 NEXT I
460 PRINT"    UDSKRIFT AF STORE     KARAKTERER PAA        PRINTER.
470 PRINT"  INDTAST DET TEGN.     SOM PLAKATTEKSTEN     SKAL OPBYGGES AF:";
480 POKE198,0:WAIT198,1: GET C$: PRINTC$
490 PRINT"  SKRIV TEKSTEN, DER    ØNSKES FORSTØRRET."
500 PRINT"      MAX 10 TEGN.     ":           REM * TIL SKAERM: 'MAX 2 TEGN' *
510 POKE657,0:          REM * TILLADER SKIFT MELLEM GRAFIK OG BOGSTAVER *
520 INPUT S$
530 T=LEN(S$)
540 S=NOT((PEEK(36869)AND8)/-8)
550 IF NOT S THEN CG=(PEEK(36869)AND7)*1024
560 IF S THEN CG=(PEEK(36869)AND3)*1024+32768
570 IF T>10 THEN 500:       REM * TIL SKAERM: 'IF T>2 THEN 500' *
580 FOR J= 0 TO 7:       REM * 8 LINIER *
590 :  PRINT#1,CHR$(8):PRINT#1,CHR$(15);:     REM * REDUCERET LINIESKIFT *
600 :  FOR P= 1 TO T:  REM * ANTAL TEGN *
610 :    A=ASC(MID$(S$,P,1))
620 :    IF (31<A)  AND (A<64)  THEN 660       FORTSAET
630 :    IF (63<A)  AND (A<96)  THEN A=A-64:   GOTO660
640 :    IF (159<A) AND (A<192) THEN A=A-64:   GOTO660
650 :    IF (191<A) AND (A<224) THEN A=A-128:  GOTO660
660 :    K=PEEK(CG+A*8+J)
670 :    FOR I=7 TO 0 STEP-1:  REM * 8 PIXELS PR. BYTE *
680 :      IF K AND B(I) THEN PRINT#1,C$;:NEXT I:GOTO710
690 :      PRINT#1," ";
700 :    NEXT I
710 :  NEXT P
720 NEXT J: PRINT
730 PRINT" ØNSKES FLERE TEKSTER  UDSKREVET? (J/N)"
740 GET A$: IF A$="J" THEN 470
750 IF A$="N" THEN PRINT#1: CLOSE1:END
760 GOTO 740
```

# Program, der får printeren til at skrive med kæmpe-bogstaver

Christian Dirksen har sendt os dette program, der forstørrer skriften til printeren, så hvert enkelt tegn bliver otte gange højere og otte gange bredere end normalt. En god idé til overskrifter og lignende.

Programlistningen indeholder alle oplysninger om strukturen - en kompliment til Christian Dirksen for den meget overskuelige præsentation...

## VIC-kontakten

Navn: ____________

Adresse: ____________

____________

Postnr.: ____________

BY: ____________

Tlf.: ____________

Bemærkninger: ____________

____________

____________

____________

____________

Kuponen sendes til VI & VIC, Kornager 29, 7100 Vejle.

# VIC programmer

Lad andre få mulighed for at købe de programmer, du udvikler til VIC.

Rubrikannoncer her i bladet koster 1 kr. pr. spaltemillimeter, og for overskuelighedens skyld bør annoncerne bygges op på samme måde, som det kan ses her på siden. Det vil sige, at annonceteksten skal indeholde oplysninger om nødvendigt udstyr, hvilket medium, progammet findes på (kassettebånd, disk eller cartridge), prisen - og naturligvis oplysninger om, hvor man kan købe programmet.

Brug kuponen til din annonce.

## VIC program sælges

**Programmets navn:** ____________

**Beskrivelse:** ____________

____________

____________

____________

____________

**Hardware:** ____________ **Medium:** ____________

**Pris:** ____________

**Købes hos:** ____________

____________

Kuponen sendes til VI & VIC, Kornager29, 7100 Vejle.

# Lad VI & VIC hjælpe dig

**VI & VIC skal være med til at du kan få mere fornøjelse af din VIC-20, og derfor indbyder vi dig til at komme med dine praktiske eller mere teoretiske spørgsmål.**

**Pladsen vil naturligvis sætte grænserne for, hvor mange spørgsmål, der kan blive besvaret her i bladet, men vi vil naturligvis først og fremmest lægge vægt på at besvare de spørgsmål, som optager mange VIC-brugere.**

Skriv til VI & VIC på adressen Kornager 29, 7100 Vejle, hvis du har spørgsmål, du gerne vil have behandlet.




Ikke så underligt, for ingen andre tilbyder så meget computer til så små priser - se godt efter priserne i den nærmeste fremtid.

Commodore computerne er »stærke sager« - se selv:

**COMMODORE 64**
»Rigtigt tastatur«.
64K RAM indbygget hukommelse.
8 sprite-planer. 3-D virkning.
40 karakterer og 25 linier.
16 tastaturstyrede farver.
Avanceret Hi-Fi synthesizer.
Kan kobles til stereo-anlægget.

**VIC 20**
Skrivemaskinetastatur med 8 funktionstaster
- kan omstilles til grafik og tekst.
5K RAM - kan udvides til 32K RAM.
Farver i 255 kombinationer.
Lyd: 3 tonegeneratorer til musik + 1 lydgenerator.

Der er mange stærke programmer til både seriøse formål og til en masse elektronisk »spøg & skæmt«. Og der kommer nye til hver dag.

Der er en lang række tilslutningsmuligheder og tilbehørsprogrammet er helt i top.

Få dig en »prøvetur« hos din lokale Commodore forhandler.

commodore
COMPUTER

**- hva' ellers !**